

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computing device comprising a processor, and a process scheduling apparatus for performing parallel processing of a plurality of processes respectively having assigned priorities, said process scheduling apparatus comprising:

at least one delayed task processing unit for executing delayed tasks among the plurality of processes, the at least one delayed task processing unit having a queuing table in which the delayed tasks are to be registered and having an assigned priority that is variable, wherein the delayed tasks include a task involved in an interrupt handler task for which processing can be delayed;

a plurality of normal process executing units for executing one of the plurality of processes other than the delayed tasks, and having an assigned priority identical to the priority of the executed process;

a process scheduling unit for sequentially activating the at least one delayed task processing unit and the normal process executing units according to the priorities assigned to the at least one delayed task processing unit and the normal process executing units to make the at least one delayed task processing unit and the normal process executing units execute corresponding processes;

a delayed task registration processor for registering a newly generated delayed task and a priority thereof in the queuing table of the at least one delayed task processing unit;

a delayed task priority controller for selecting the delayed task of highest priority from the delayed tasks registered in the queuing table; and

a process priority controller for setting the priority of the delayed task processing unit identical to the priority of the delayed task selected by the delayed task priority controller, wherein when ~~the~~ a newly generated delayed task having a priority higher than that of a currently executed delayed task is generated, the process priority controller sets the priority of the at least one delayed task processing unit ~~before an initiation of a next process following a currently executed process~~ before termination of the currently executed delayed task by suspending the currently executed ~~process~~ delayed task, setting the priority of the delayed task processing unit to the priority of the newly generated

delayed task, and then, prior to executing the newly generated delayed task, resuming the currently executed ~~process~~delayed task to completion.

2-6. (Canceled)

7. (Currently Amended) A process scheduling method for performing parallel processing of a plurality of processes respectively having assigned priorities, comprising:

sequentially executing a delayed task handling process for processing delayed tasks and normal processes for executing processes other than the delayed tasks according to priorities respectively assigned to the delayed task handling process and the normal processes, wherein the delayed tasks include a task involved in an interrupt handler task for which processing can be delayed;

registering a newly generated delayed task and a priority assigned thereto in a queuing table;

selecting a delayed task of highest priority from the delayed tasks registered in the queuing table; and

setting the priority of the delayed task handling process identical to the priority of the selected delayed task, wherein when ~~the a~~ newly generated delayed task having a priority higher than that of a currently executed delayed task is generated-generated, the priority of the delayed task handling process is set ~~before an initiation of a next process following a currently executed process~~before termination of the currently executed delayed task, and said setting the priority of the delayed task handling process comprises:

suspending the currently executed ~~process~~delayed task;

setting the priority of the delayed task handling process to the priority of the newly generated delayed task; and

resuming, prior to executing the newly generated delayed task, the currently executed ~~process~~delayed task to completion.

8-14. (Canceled)

15. (Currently Amended) A computer program recorded on a computer-readable storage medium for performing parallel processing of a plurality of processes respectively having assigned priorities, wherein said computer program includes:

at least one delayed task processing unit for executing delayed tasks among the plurality of processes, the at least one delayed task processing unit having a queuing table in which the delayed tasks are to be registered and having an assigned priority that is variable, wherein the delayed tasks include a task involved in an interrupt handler task for which processing can be delayed;

a plurality of normal process executing units for executing one of the plurality of processes other than the delayed tasks, and having an assigned priority identical to the priority of the executed process;

a process scheduling unit for sequentially activating the at least one delayed task processing unit and the normal process executing units according to the priorities assigned to the at least one delayed task processing unit and the normal process executing units to make the at least one delayed task processing unit and the normal process executing units execute corresponding processes;

a delayed task registration processor for registering a newly generated delayed task and a priority thereof in the queuing table of the at least one delayed task processing unit;

a delayed task priority controller for selecting the delayed task of highest priority from the delayed tasks registered in the queuing table; and

a process priority controller for setting the priority of the delayed task processing unit identical to the priority of the delayed task selected by the delayed task priority controller, wherein when ~~the~~ a newly generated delayed task having a priority higher than that of a currently executed delayed task is generated, the process priority controller sets the priority of the at least one delayed task processing unit ~~before an initiation of a next process following a currently executed process~~ before termination of the currently executed delayed task by suspending the currently executed ~~process~~ delayed task, setting the priority of the delayed task processing unit to the priority of the newly generated delayed task, and then, prior to executing the newly generated delayed task, resuming the currently executed ~~process~~ delayed task to completion.